AD-A095 693 INCO INC MCLEAN VA F/G 17/2

STANDARD SOFTWARE BASE RELEASE III.(U)

JAN 81 G MORGAN F30602-79-C-0262

UNCLASSIFIED INCO/1138A-1080-TR-47-D/F RADC-TR-80-376 NL

1 OF 1
AD
A095693

END
DATE
FILMED
3-81
DTIC

LEVEL II

12

RADC-TR-80-376
Final Technical Report
January 1981

AD A095693

STANDARD SOFTWARE BASE RELEASE III

INCO, Inc.

Gary Morgan

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-80-376 has been reviewed and is approved for publication.

APPROVED: Fred Haritatos

FRED HARITATOS
Project Engineer

APPROVED: Owen R. Lawter

OWEN R. LAWTER, Colonel, USAF
Chief, Intelligence and Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRDE) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 19 RADC-TR-80-376	2. GOVT ACCESSION NO. AD-A095693	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) STANDARD SOFTWARE BASE RELEASE III.		5. TYPE OF REPORT & PERIOD COVERED 9 Final Technical Report, 24 Sep 79 — 25 Sep 80,
		6. PERFORMING ORG. REPORT NUMBER 14 INCO/1138A-1080-TR-47-D/F
7. AUTHOR(s) Gary Morgan		8. CONTRACT OR GRANT NUMBER(s) F30602-79-C-0262
9. PERFORMING ORGANIZATION NAME AND ADDRESS INCO, Inc. 7916 Westpark Drive McLean VA 22102		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 31025F 21830156 17 01
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDE) Griffiss AFB NY 13441		12. REPORT DATE 11 January 1981
		13. NUMBER OF PAGES 49 12 41
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Fred Haritatos (IRDE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

AN/GYQ-21(V)
Standard Software Base
Terminal Transparent Display Language
Gateway Programs
AUTODIN
Message Support System
Queue Manager Concept

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the work performed to expand the capabilities of the USAF Standard Software Base (SSB) supporting AN/GYQ-21(V) operations. The report discusses maintenance and improvements made to the baseline system, and the development and implementation of advanced capabilities and services. These improvements include refined interfaces with the AUTODIN network improvements to internal message routing and control, expansion of message storage capacities, improved user operating procedures and i

(Cont'd)

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Item 20 (Cont'd)

preliminary interface capability with the Message Support System. ~~(MSS)~~. Development efforts to improve baseline SSB capabilities include the introduction of a revised queue management structure, a new user function (REVIEW), and expansion of function-key operating procedures.

TABLE OF CONTENTS

EVALUATION

The Standard Software Base (SSB) was developed in an effort to supply a common system software for AN/GYQ-21(V) communications processing capabilities, data management services, applications programs with which to access and control message traffic, and enhancements to the AN/GYQ-21(V) operating systems.

Under this effort, the integration of applicable Defense Special Security Communications Systems (DSSCS) and Message Support Subsystem (MSS) software into the SSB was addressed as well as a continuation of the analysis, design, and implementation of the enhancements to the baseline SSB capabilities.

A key achievement of the program was the integration of the SSB and the MSS which provided the users access to full-text searching, automatic message dissemination and message retrieval capabilities.

A significant enhancement to the system was the introduction of a Review function which provides the SSB with the capability to review, print, transfer and delete messages while they are displayed at a terminal without the necessity for exiting one function in order to call up another to perform such actions. This enhancement results in a faster and more efficient management of message traffic flow.

This effort was completed under RADC Technical Planning Objective P1E.

Fred Haritatos

FRED HARITATOS
Project Engineer

SECTION 1. INTRODUCTION

1.1 Purpose. This report describes work performed for the Air Force Intelligence Service under the provisions of Rome Air Development Center Contract No. F30602-79-C-0262. The report discusses the work accomplished and identifies the documentation provided to the government during the period 25 September 1979 to 24 September 1980.

1.2 Background. The Standard Software Base (SSB) is a major component of the United States Air Force Common User's Baseline for the Intelligence Community (CUBIC). The SSB is the result of intelligence data handling developments initiated by the Air Force Intelligence Service (AFIS/IND) in 1975 to help users of the AN/GYQ-21(V) to develop their individual computer systems. Development of the SSB has proceeded under a carefully managed program to provide network and communications processing capabilities, data management services, applications programs with which to access and control message traffic, and enhancements to the AN/GYQ-21(V) operating systems. Figure 1 illustrates a typical SSB Release III system configuration.

1.3 Objectives. The primary objectives of work performed under the contract was to extend the USAF Standard Software Base (SSB) supporting AN/GYQ-21(V) operations. The following areas of interest were addressed to meet this primary objective. They are:

1) The integration of applicable Defense Special Security Communications System (DSSCS), and Message Support Subsystem (MSS) software into the SSB.

2) Analysis and design of distributed data base concepts and the use of High Speed Search Technology (HSST) to enhance DODIIS data flow requirements; and continued analysis, design, and implementation of enhancements to baseline SSB capabilities.

3) Configuration control procedures, SSB distribution services and training support facilities.

4) Configuring for site installation, and implementation of SSB capabilities for potential SSB users.

5) Providing technical documentation for the SSB capabilities provided under the contractual effort.

1.4 Report Organization. This final technical report consists of three sections. Section 1 provides background information; technical achievements are discussed in Section 2; and the documentation provided the government is identified in Section 3.

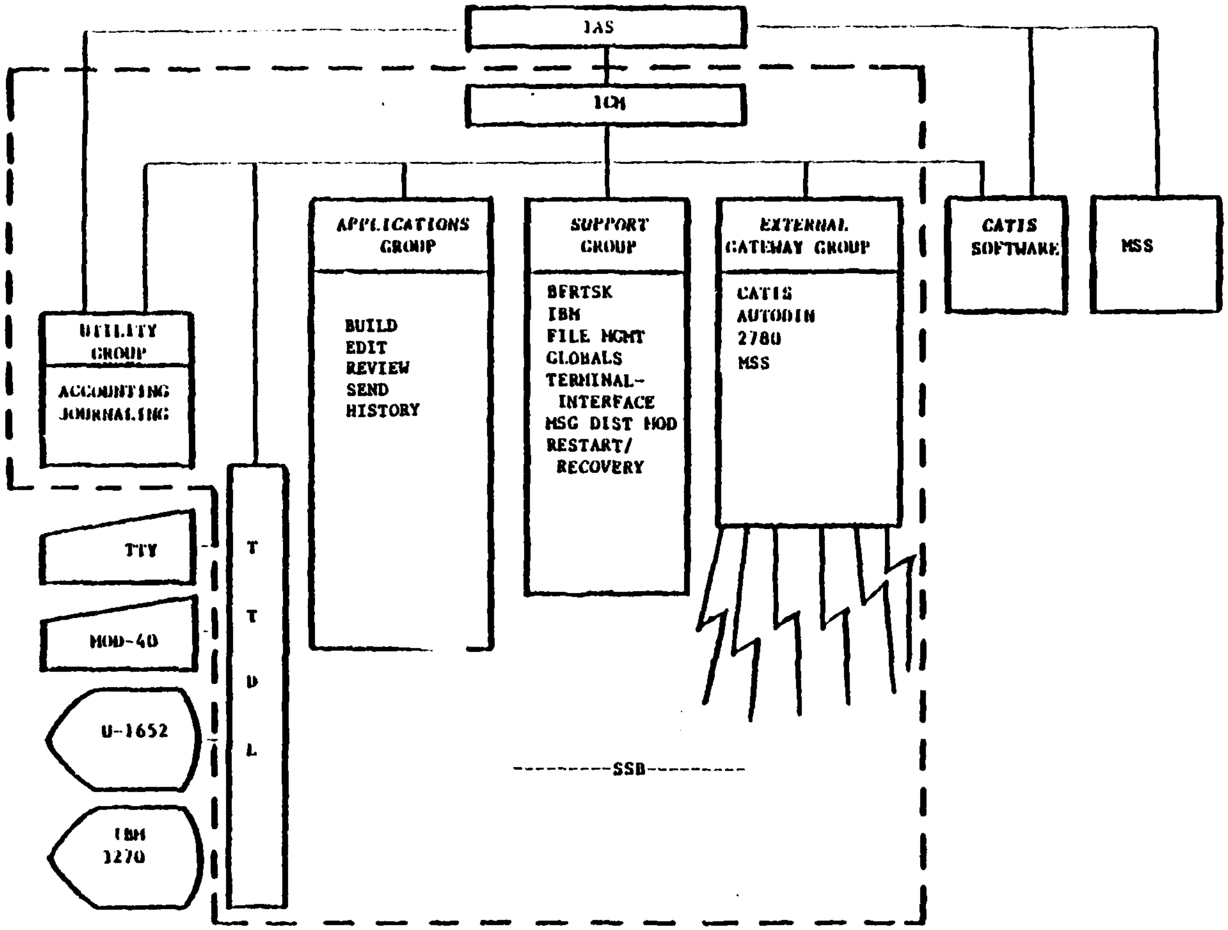


Figure 1. SSB RELEASE 3.1 SYSTEM

SECTION 2. Technical Accomplishments.

2.1 SSB/DSSCS Interface. Enhancements to the SSB and CSP AUTODIN Gateways were implemented and tested with the CUBIC Communications Support processor (CSP) to provide for improved error recovery. A new version of the CSP system was generated and successfully placed into operation. Modifications to the original CSP software included security table changes requested by site personnel. A draft of the SSB/CSP Test and Implementation Plan was delivered to AFIS/IND in November 1979 for review and approval.

Between November - December 1979, Mode II interface software on the CSP was reconfigured to support eight communications lines. Message traffic was successfully passed between the CSP and four TTY-Mod 40s. Several communications hardware problems were identified in connecting the lines to the CSP system. During these preliminary testing activities, a need became apparent for software to support a hardware signal requiring synchronization of encrypted data prior to message transmission. The software will be tested after new hardware ROMs are installed in the BR1569 multiplexor.

In January 1980, the CSP was installed at TCATA (Ft. Hood, Texas) and integrated with the extent SSB system there. Mode II software was modified to perform cryptographic resynchronization prior to message transmission.

Maintenance work was performed on the AUTODIN Receive Gateway to correct a problem reported by USAFE concerning the inability of the system to receive FLASH precedence traffic due to the rejection of the select character. Additional development took place to increase the size of the message file handler, TISFIL from 32K blocks to 64K blocks.

2.2 SSB/GENSER Interface. The design and coding for the automatic interface to transmit GENSER traffic was completed. Early in the design phase it was determined that SSB would interface with the AUTODIN system via an Analytics Telecommunications Line Controller (TLC). Systems using this interface to the AUTODIN/DSSCS system have been successfully implemented at several sites. As a result of this decision, the SSB interface to the AUTODIN/DSSCS system would rely on proven technology to achieve a low-risk implementation. The SSB/AUTODIN Receive Gateway which received both DSSCS and GENSER traffic was modified to receive and to route GENSER traffic only. All DSSCS traffic is now routed to the service message station, printed on the system line printer. Advisory and appropriate error messages are displayed on the system console. The system queues and routine mechanisms within SSB were modified to accommodate the GENSER gateway.

2.3 SSB/MSS Interface. The integration of SSB and the Message Support Subsystem (MSS) provides SSB users with access to the full text searching, automatic message dissemination, and message retrieval

capabilities of the MSS. These capabilities provide SSB users with the following enhancements:

a. Automatic dissemination of incoming DSSCS/GENSER messages into SSB-supported review queues based on site-defined criteria specified in profiles.

b. Message retrieval based on a variety of indexing keys.

c. MSS-supported capabilities for on-site profile maintenance.

2.3.1 Overview and Objectives. The overall data flow in an SSB/MSS integrated system is shown in Figure 2. In this integrated system, MSS provides a package of capabilities based on automatic text profiling, while SSB provides a DSSCS/GENSER communications interface, a package of user-oriented message handling programs, and flexible terminal support software (TTDL).

The objective of the integration of MSS into SSB was to perform the following data transfers without loss of data in the event of system crashes:

a. Incoming messages must be transferred from the SSB message file to the MSS message file.

b. MSS-generated profile data and dissemination information must be transferred from the MSS message file to the SSB queue file.

c. Queries specified at an SSB user terminal must be transferred to the MSS query processor.

d. Query responses must be transferred from MSS to the SSB queues to provide user access to the query response.

2.3.2 Software Interface Design. The interface beween the SSB system and the MSS system is designed to satisfy the following constraints:

a. Preservation of intra-system autonomy to the fullest extent so that future intra-system changes would not affect the interface.

b. Limiting the number of interaction points between the two systems. Specifically, all requests for MSS services from SSB are made by a single SSB module (MSSGTY), and all requests for SSB services from MSS are made from a single MSS module (MSSINT).

c. IAS File Control Services (FCS) are used to transfer data aggregates. All disk files created and written by SSB tasks are "read-only" for MSS tasks, and all disk files created and written by MSS tasks are "read-only" for SSB tasks, except for query files created by MSS which are deleted by SSB when the query is answered.

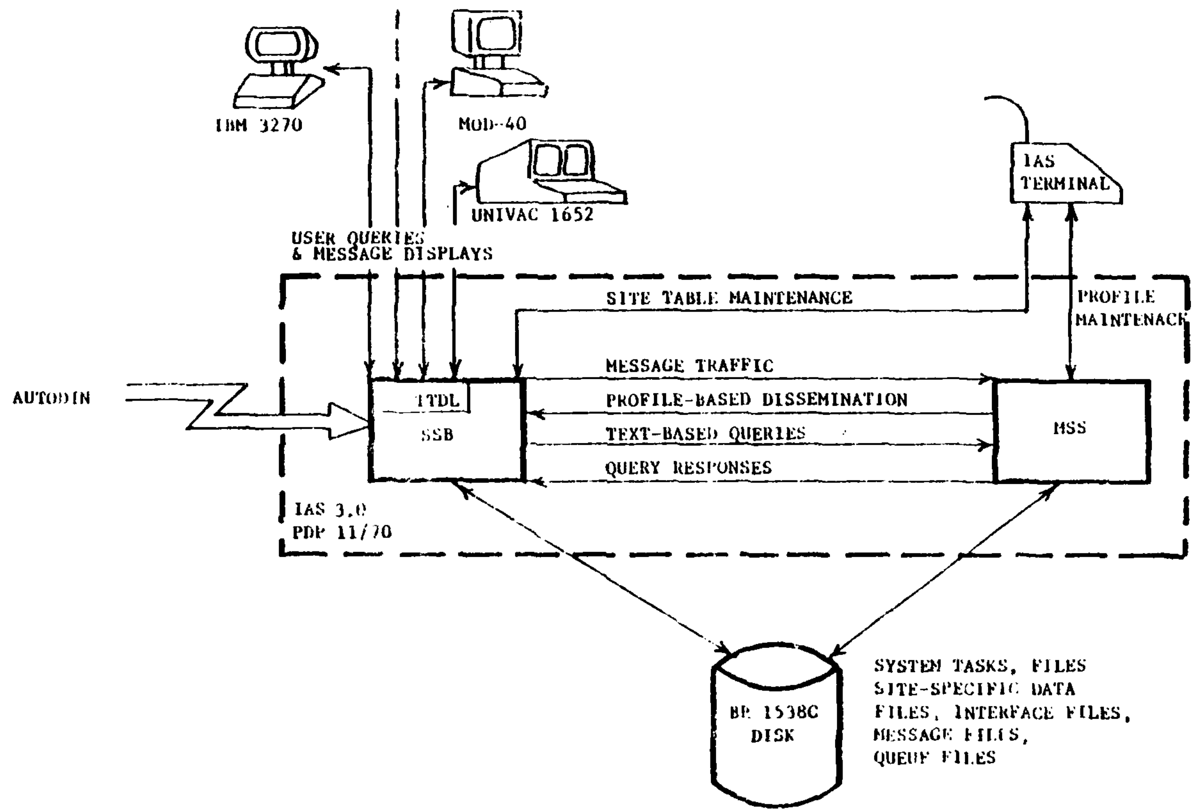


Figure 2 SSB/MSS Data Flow

d. All intra-SSB service requests utilize 16 word Service Request Blocks (SRB), while all inter-system and intra-MSS service requests utilize IAS-supported inter-task variable-data-block (VSDR) directives.

e. All IAS directives used are compatible with RSX 11D, unless explicitly noted in the constraints section of the program specifications.

2.3.3 Pertinent Software Modules. The SSB system maintains a single interface gateway (MSSGTY) system. MSSGTY transmits and receives all data communications between the two systems. MSSGTY uses ICMMU directives for intra-SSB communications, and IAS directives for commuunication to the MSS interface modules.

MSS maintains an interface program called MSSINT. MSSINT handles all traffic to and received from the SSB system through the MSS Gateway. MSSINT communicates via IAS directives and accepts only one routed message and one query at a time, to avoid any requirement for redundant queue management. Subsequent communications between the two systems depend upon MSSGTY reception of the proper acknowledgements. The other modules comprising the SSB/MSS interface include:

TISMDM - Receives incoming message traffic from the AUTODIN receive gateway, TISGTA. Routes appropriate message traffic to SBGMSS.

SBGMSS - Translates a TCF message to JANAP format and writes the data to an FCS file.

SBGDAN - Receives dissemination requests through MSSGTY and distributes messages to specified user review queues.

OSIPRE - Message preprocessor

...FLM - The file manager

DISSEM - Dissemination program

QRYFIL - Builds QUERY.QRY File from analyst input

QRYPRC - Builds query record, parsing the QUERY.QRY and converting any logic statements to reverse polish notation.

...IXU - Notation - Index cross reference

...QRC - The query resolver

QRYOUT - Builds the QUERY.ANS from QUERY.QRY and the answer record.

INQRY - Checks format and does preliminary validation for queries.

2.3.4 Data Structures. The basic data structures involved in the SSB/MSS interface are as follows:

MSGBLK - An 8-word structure used for intertask service requests between the SSB interface handler (MSSGTY) and the MSS interface handler (MSSINT).

MSSSB.MSG-An FCS file containing the JANAP equivalent of a received AUTODIN message, with all control characters (STX, EM, ETX) removed

QUERY.QRY-An FCS file containing the fields specified by the user via the query entry module (QRYFIL).

QUERY.ANS-An FCS file containing a copy of the user query together with a list of MSN/DTG pairs corresponding to the messages which satisfied the query.

SRB - A 16-word structure used for intertask service requests between SSB modules.

The format of these structures is shown and described in Figures 3 through 8.

2.3.5 Start-Up Procedures. Each system will be restarted independently using the appropriate start-up command files. MSS may be started via cold start or warm start command files. Cold start procedures imply all MSS files will be result and reinitialized. Warm start procedures will allow preservation of message files and a continuation of processing from existing files. The various start-up possibilities are shown in Figure 9.

The SSB restart is achieved via an indirect-MCR command file called SSBT3.IND. This command file will include a question as follows:

DO YOU WANT MSS [Y/N]?

If the operator specifies "Y" then SSBT3.IND will invoke the command file for an MSS warm start. This question will be bypassed and will default to "N" if SSBT3.IND detects MSS has been restarted separately. In all cases the step of SSBT3.IND will be to run TISINT which will call MSSGTY to ensure a directive to MSSINT containing a "start-up" function code. MSS will be treated as inactive and no further directives will be issued if MSSINT does not respond to the "start-up".

The interactions involved in a SSB/MSS start-up are depicted in Figure 10 for an SSB initiated start-up. In the case where MSS does not

Offset Name	Offset Byte			Offset Byte	Offset Name
	1	Function Code		0	MSFC.
	3	Action Code		2	MSACT.
MSERI.	5	Error Code	Status	4	MSST.
	7	File Version Number		6	MSID.
	11	Message DAN		10	MSDAN.
	13			12	
	15	Priority		14	MSPRI
	17	MSN		16	MSMSN

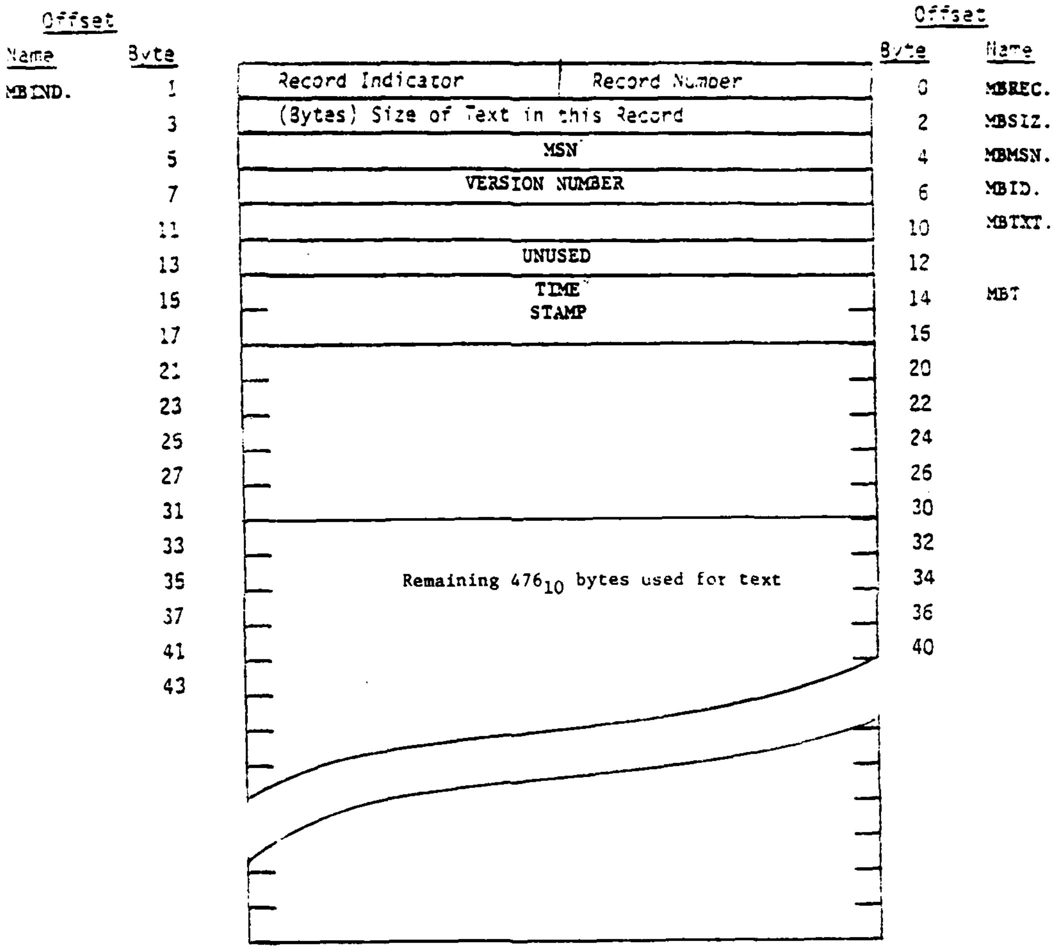


Figure 4. MSSSSB.MSG Format

	Byte	Name
SUBJECT AREA CODE	0	QRSAC
NOT USED / TERMINAL ID	6	QRTID
QUERY NAME	10	QRNAM
REFERENCE CODE	24	QRRTC
RETURN TERMINAL ID	32	QRRID
COMMENTS	34	QRCOM
MAXIMUM NUMBER OF RESPONSES (ASCII)	104	QRMAX
NOT USED	110	QRBDT
ENDING DATE-TIME GROUP	124	QREDT
ORIGINATOR(S)	140	QRORG
LOGICAL QUERY STATEMENT	224	QRLQS
	776	

Figure 5. QUERY.QRY. Block 1 of 2

	OCTAL OFFSET	
	Byte	**Name**
	0	
QUERY NAME (ASCII)	22	QINAM
UNUSED	36	
LOGIC MODE SWITCH (CONSTANT FOR SSB)	54	QIMOD
MAXIMUM NUMBER OF RESPONSES	56	QIMAX
UNUSED		
BEGINNING DTG (BINARY)	66	QIBDT
ENDING DTG (BINARY)	72	QIEDT
QUERY ORIGINATOR COUNT	76	QIORC
KEYWORD COUNT INDEX	100	QIKCX
STATEMENT BYTE COUNT INDEX	102	QISCX
REFERENCE COUNT (CONSTANT FOR SSB)	104	QIRFC
RETURN SUBJECT AREA	106	QIRTN
UNUSED | RETURN TERMINAL ID		
REVERSE POLISH STACK AND INDEX KEY NODES (VARIABLE LENGTHS)	120	
	776	

Figure 6. QUERY.QRY. BLock 2 of 2

	OCTAL OFFSET	
	Byte	Name
SUBJECT AREA CODE	0	ANSAC
TERMINAL ID	6	ANTID
QUERY NAME	10	ANNAM
RETURNEE CODE	24	ANRTC
RETURN TERMINAL ID	32	ANRID
COMMENTS	34	ANCOM
MAXIMUM NUMBER OF RESPONSES	104	ANMAX
BEGINNING DATE-TIME GROUP (ASCII)	110	ANBDT
ENDING DATE-TIME GROUP (ASCII)	124	ANEDT
ORIGINATORS	140	ANORG
LOGICAL QUERY STATEMENT	224	ANLQS
	270	
	776	

Figure [illegible] QUERY ANS, Block 1 of 2

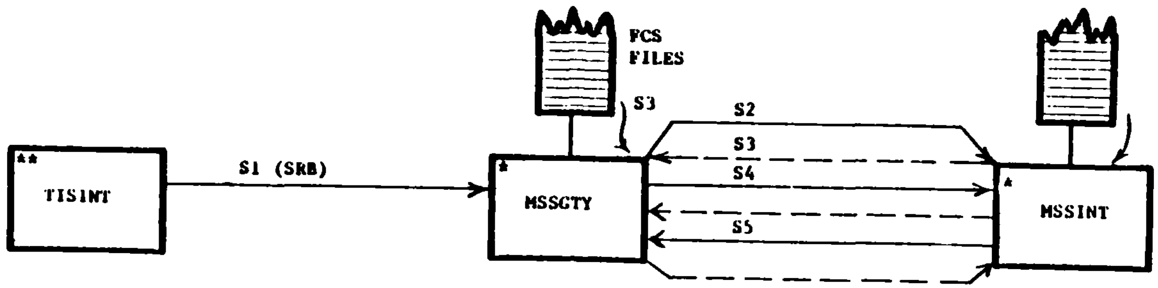
Field	Byte	Name
QUERY REQUEST CODE (TQR DAN)	0	ANTQR.
	2	
HIT COUNT (BINARY)	4	ANHTC.
NUMBER OF MESSAGES RETURNED	6	ANCNT.
MESSAGE SEQUENCE NUMBER	10	ANMSN.
BINARY DATE-TIME-GROUP MSG(1)	12	ANDTG.
	14	
MSG(2)	16	
	20	
	22	
	24	
MSG(3)	26	
	30	
	32	
MSG(4)	34	
	36	
	40	
MSG(5)	42	
	44	
	46	
MSG(6)	50	
	52	
	54	
	56	
MSG(7)	60	
	62	
	64	
MSG(8)	66	
	774	
MSG(N)	776	

Figure 8. QUERY.ANS, Block 2 of 2

	SSB UP	SSB NOT UP
MSS UP	NO ACTION	USE SSBT3.IND for SSB START-UP BUT DO NOT ASK FOR MSS. TISINT WILL CALL MSSINT VIA MSSGTY.
MSS NOT UP	MSS BROUGHT UP VIA 1. Warm Start 2. Cold Start (IF MSS ALREADY UP, COMMAND FILE SHOULD NOT ALLOW RESTART. CHECK WILL BE MADE).	1. SSBT3.IND WILL ASK ...WANT MSS? (Y/N) TISINT will call MSSINT THROUGH MSSGTY. CALL FAILURE HALTS MSS TRAFFIC UNTIL MSS calls MSSGTY. 2. MSS BROUGHT UP SEPARATELY VIA COLD START OR WARM START.

Figure 9. INTERRELATION BETWEEN SSB RESTART AND MSS WARM-START/COLD-START.

START-UP FROM SSB



S1. TISINT sends SRB to MSSCTY with start-up function code.

S2. MSSCTY sends start-up signal to MSSINT.

S3. Return ACK is sent by MSSINT.

S4. If MSSINT responds successfully to the start-up request, MSSCTY restarts from the existing FCS files and receives the appropriate ACKs.

S5. MSSINT contacts restart programs within MSS which may result in requests to SSB.

*SOFTWARE UNDER DEVELOPMENT
**MODIFIED SOFTWARE

Figure 10. INTERACTIONS INVOLVED IN INTEGRATED SYSTEM START-UP INITIATED BY SSB

respond to the start-up requst, FCS files will continue to accumulate, but will not be forwarded to MSSINT until such time as MSS sends a start-up request or responds to a subsequent restart request. The same comments apply to MSS-initiated restart.

2.3.6 SSB/MSS Message Traffic Interface. The SSB/MSS Message Traffic Interface is depicted in Figure 11.

The Standard SSB AUTODIN Receive Gateway, TISGTA, receives and parses incoming data. The SSB message routing tables were modified to permit routing to MSS using standard SSB routing tokens. This permits TISMDM, the SSB message distribution module, to send an SRB to the file translation module, SBGMSS. It is optional whether the message is made available to SSB users at this point or whether availability is deferred until MSS determined the dissemination.

SBGMSS opens the TCF message and converts it to JANAP format, record by record. In the TCF-to-JANAP conversion, the TCF end-of-record marks are converted to <CR> <CR> <LF>, the AUTODIN end-of-record marks. SBGMSS then writes the message to an FCS file in blocks. SBGMSS acknowledges message receipt to TLISMDM and routes the message to MSSGTY, the SSB/MSS Gateway.

Upon SRB receipt, MSSGTY routes the message to MSSINT (the MSS interface control program) using the eight-word message data block (MSGBLK) (see Figure 3).

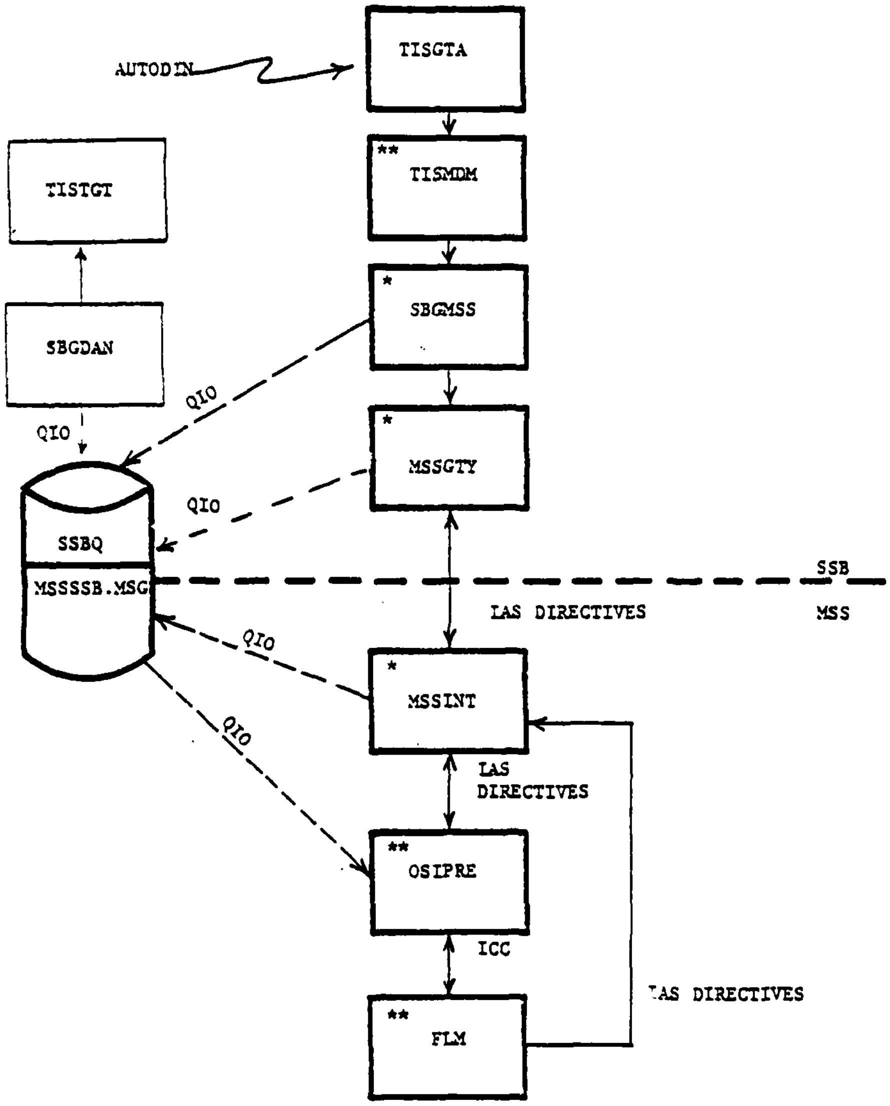
Once MSS has received the message from MSSGTY, MSS program modules perform a full text search of the message. The message is matched against user-defined interest profiles, resulting in a list of interested subject areas.

MSS issues a request for dissemination to MSSGTY. MSSGTY receives the request and routes it to the SSB dissemination modules, SBGDAN. SBGDAN accesses the MSN and subject area list and distribution the message to the proper subject areas.

2.3.7 Interactions for Message Traffic. The software interactions for mesage traffic are divided into two processes: the incoming message traffic interface and the request for dissemination. Figure 12 shows the interactions for incoming messages and Figure 13 shows the interactions for message dissemination.

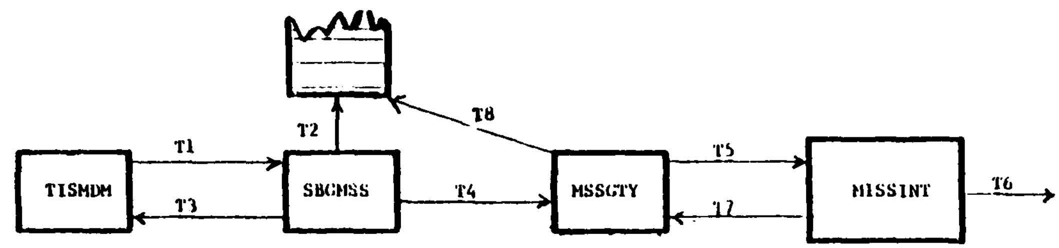
2.3.8 SSB/MSS System Functions.

2.3.8.1 SSB Software. SBGMSS is responsible for receiving messages routed to MSS and converting the message text to JANAP format stored in block format in an FCS file, MSSSB (see Figure 4). The type of the file will be ROU, PRI, IMM, or FLS depending on the message priority. The data is built in block mode. All ASCII characters appearing in the message including the AUTODIN line terminator sequence: `CR,` `CR`,



*
**MODIFIED SOFTWARE

Figure 11. SSB/MSS MESSAGE TRAFFIC INTERFACE



T1 - TISMDM receives message with NET.MS on a destination token. Sends SRB to SBGMSS.

T2 - SBGMSS converts the TCF message to an FCS file.

T3 - SBGMSS ACKs and releases TISMDM

T4 - SBGMSS routes SRB to MSSGTY for message routing.

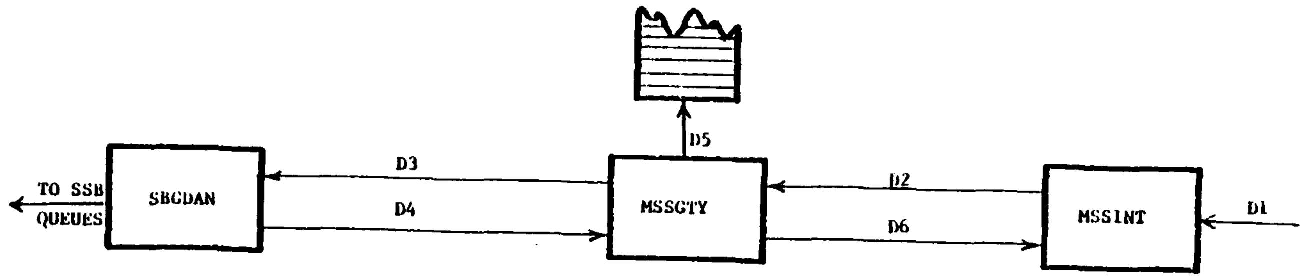
T5 - MSSGTY routes message to MSSINT via a VSDR.

T6 - MSS programs place the message on the MSS message file and enters the message on the MSS restart log.

T7 - Return of ACK implies MSS now has restart control of the message and for MSSGTY to route another message to MSS.

T8 - MSSGTY time stamps the file, disabling SSB restart and insuring delivery of the file was completed to MSS.

Figure 12. Incoming Message Traffic



D1 - MSS has a message ready for dissemination.

D2 - MSS alerts MSSGTY that the message is ready.

D3 - MSSGTY routes the dissemination to SBGDAN.

D4 - SBGDAN disseminates the message and ACKs MSSGTY when complete.

D5 - MSSGTY deletes the FCS file.

D6 - MSSGTY ACKs completion of dissemination to MSSINT.

Figure 13. Request for Message Dissemination

'LF,' are stored in the file. The block size is 512. bytes and the message text may cross block boundaries. A header of up to 30 characters is included in each file block, therefore, the maximum number of text characters per block is 476. SBGMSS then routes the message data to MSSGTY for routing to MSS.

2.3.8.2 MSS Software. MSSINT is the only MSS program interacting with MSSGTY. MSSINT is installed, but not active until invoked via an IAS system directive from MSSGTY. MSSINT handles the protocol between the two programs and passes required information to OSIPRE, and MSS preprocessor, via IAS system directives.

OSIPRE was modified to be activated via a directive call from MSSINT. Its interface to the MSS concentrator element was removed. OSIPRE accesses the specified version of MSSSB.msg directly via IAS supported I/O directives until the entire file is read and processed. OSIPRE then sends a directive to MSSINT. Error conditions are reported in this directive when they arrive.

MSSINT communicates general message status to MSSGTY. Status information is reported to MSSINT from two sources. OSIPRE returns an ACK to MSSINT when message processing is finished and the message has been written to the MSS file. MSSINT then sends an ACK to MSSGTY. The File Manager Module (FLM) sends a directive to MSSINT when routing assignments are complete. MSSINT forwards the status information to MSSGTY. MSSINT also reports error conditions to the MSSERR task in the form of error codes. These codes are translated into appropriate error message strings and printed out on the system console.

2.3.9 Query Processing. As in message input, MSSGTY interact only with MSSINT during query processing. Query processing flow is illustrated in Figure . The function of the query processing programs is described in the following paragraphs.

Program	System	Author	Function
QRYPRC	MSS	OSI	Transmits query via ICC pseudo-handler
QRYANS	SSB	INCO	Translate query responses from MSS to display format
QRYFIL	SSB	OSI	Convert data from analyst's query to internal MSS processing format
QRYOUT	MSS	OSI	Send responses to MSSINT
INQRY	SSB	OSI	Interface to analyst terminals.

The logical flow between these programs is shown in Figure 14. MSSERR is the error message display program. It routes error messagess to the system console. The following sections describe these modules and their interaction.

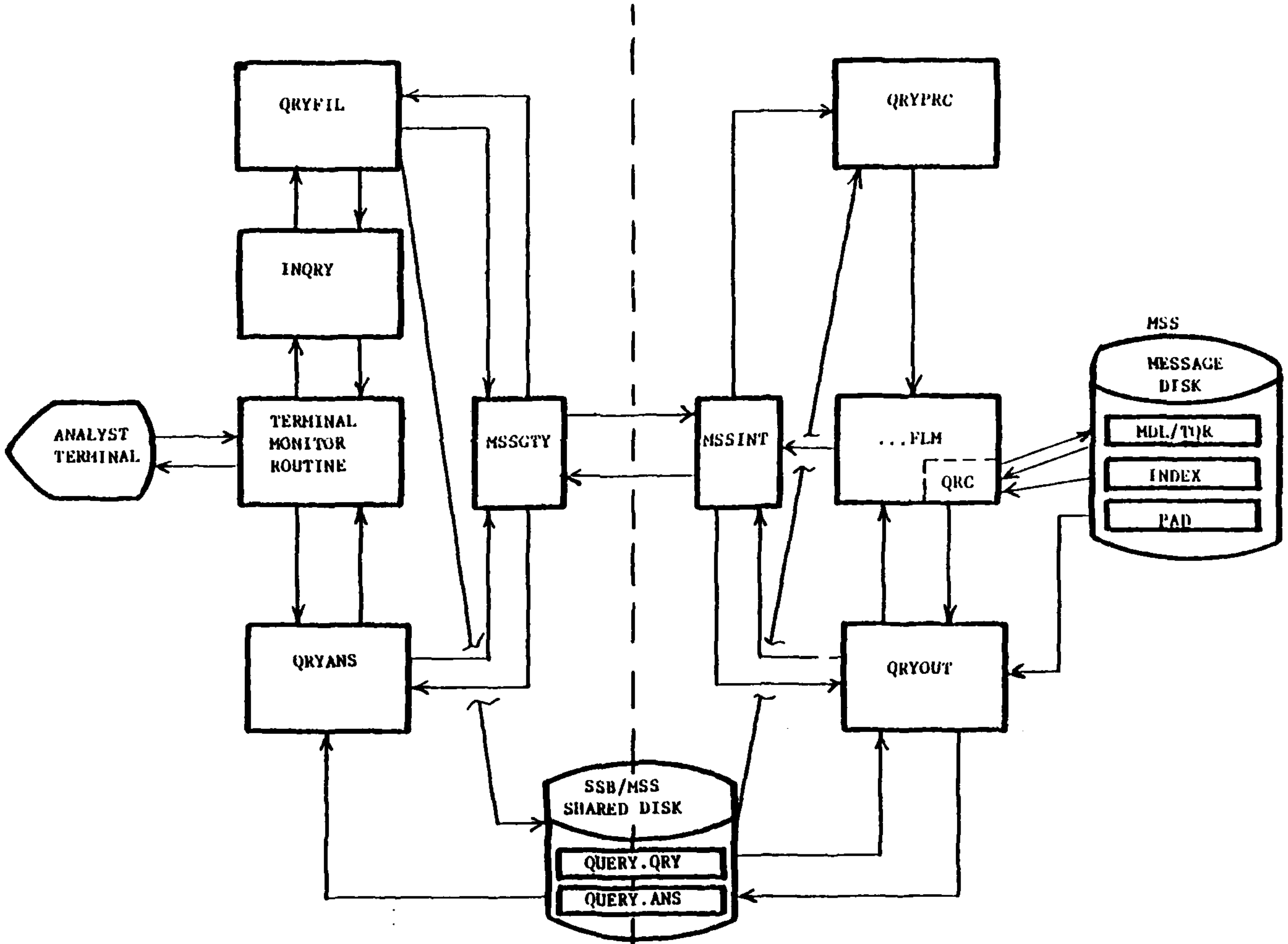


Figure 14. SSB/MSS Query Function Intertask Communication

a. INQRY Module. INQRY provides the user with a blank query form on request. It performs preliminary validation of completed query forms and presents the user with diagnostic messages for queries which fail syntax check in either the INQRY or QRYFIL task.

If a query form passes all syntax checks, INQRY informs the user of the QUERY.QRY file version number assigned to the query. The query input form is shown in Figure 15 as presented and as completed.

b. QRYFIL Module. QRYFIL is a single user task which validates the contents of the user completed query form. If syntax errors are discovered, error messages will be sent to INQRY via buffer passing to be displayed on the user screen.

When a valid query is submitted, QRYFIL will write an FCS file, QUERY.QRY n to the shared disk. The version number, n, will be assigned by FCS. QRYFIL will notify the user of the query's version number via INQRY and then send an SRB to MSSGTY for routing to MSSINT. The FCS file QUERY.QRY format is shown in Figures 5 and 6.

Block 1 of file QUERY.QRY contains the new data for the query as specified by the user.

Block 2 contains the query in MSS internal processing format. QRYFIL will parse the query logic statemennt into a complex Reverse Polish Notation stack format. The data flow in a query transaction is shown in Figure 16.

c. QRYPRC Module. QRYPRC is a single-user, and is installed but not active until invoked by a directive from MSSINT. QRYPRC then opens the version of QUERY.QRY specified in the directive and writes the data in block 2 of this file to task "...FLM" via the ICC pseudo-handler "DA...". Problems are reported via MSSERR.

d. QRYOUT Module. QRYOUT is notified when a query has been entirely processed. It then opens the specified version of the QUERY.QRY file and creates a QUERY.ANS file with the same first block and version number. The QUERY.ANS file will be appended with a list of the MSN and DTG of each message which satisfied the query. The layout of QUERY.ANS is shown in Figures 7 and 8. Figure 17 shows the data flow involved upon return of a query answer from MSS to the SSB module, QRYANS.

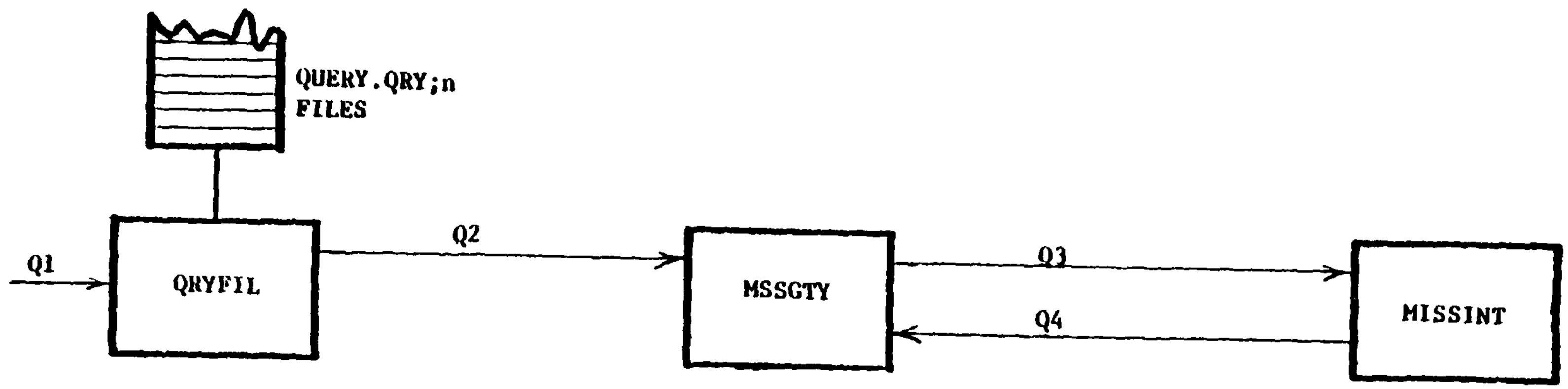
e. QRYANS Module. QRYANS is invoked by a MSGBLK from QRYOUT (via MSSGTY). QRYANS then notifies the user of the results of the query. Once the QUERY.ANS file is no longer needed, the QRYANS module deletes both the QUERY.QRY and QUERY.ANS files.

2.3.10 Profile Interface. Figure 18 shows the MSS profile software. There is no SSB/MSS interface for this function, as the subsystem will be activated via the IAS "MCR" interface from a computer terminal.

```
QUERY NAME: ____________  RETURNEES SUBJECT AREA: ______
QUERY ANSWER LIMIT: ____  REMARKS: ______________________________________

QUERY SPECIFICATION

DTG LIMITS:  FROM: ____________  TO: ____________

  ORIGINATOR CODES: ______________________________________________________
QUERY: ___________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________


53 PAGE 1 OF 1
```

```
QUERY NAME: CUBA-AFRICA_  RETURNEES SUBJECT AREA: WTCHME
QUERY ANSWER LIMIT: 25__  REMARKS: CUBA AND THE HORN OF AFRICA____________

QUERY SPECIFICATION

DTG LIMITS:  FROM: 152359ZAPR78  TO: 182359ZAPR78

  ORIGINATOR CODES: ______________________________________________________
QUERY: SUBJECT EQUALS ETHIOPIA OR SOMALIA AND SUBJECT EQUALS CUBA AND______

DISSEMINEE EQUALS MEAF OR DS14.___________________________________________

__________________________________________________________________________

__________________________________________________________________________


53 PAGE 1 OF 1
```

Figure 15. Logical Query Forms (TOP: as presented; BOTTOM: as completed)



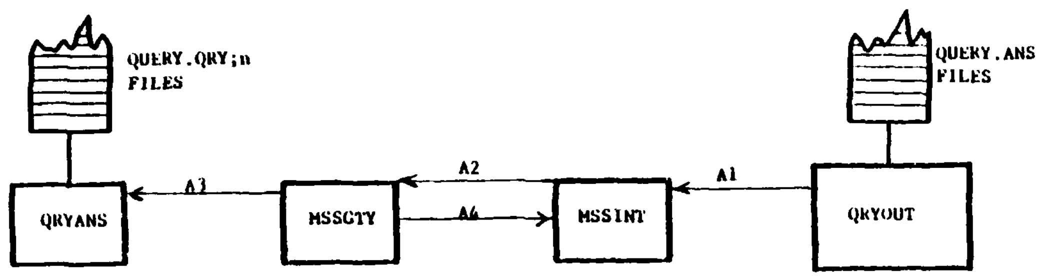
Q1 - Analyst builds a query using TTDL QRYFIL receives analyst input.

Q2 - QRYFIL writes QUERY.QRY; n file and routes the file to MSSGTY.

Q3 - MSSGTY routes QUERY to MSS.

Q4 - MSS ACKs query. MSSGTY marks query as "accepted but not completed". Another query can now be sent by MSSGTY.

Figure 16. REQUEST FOR QUERY PROCESSING



A1 - QRYOUT processes QUERY.ANS files.

A2 - MSS provides an FCS file with a list of the MSNs and DTGs. The version number of QUERY.ANS should match QUERY.QRY.

A3 - MSSCTY receives query response and sends it to QRYANS for display to the user.

A4 - MSSCTY acknowledges receipt of query answer.

Figure 17. MSS RESPONSE TO USER QUERY

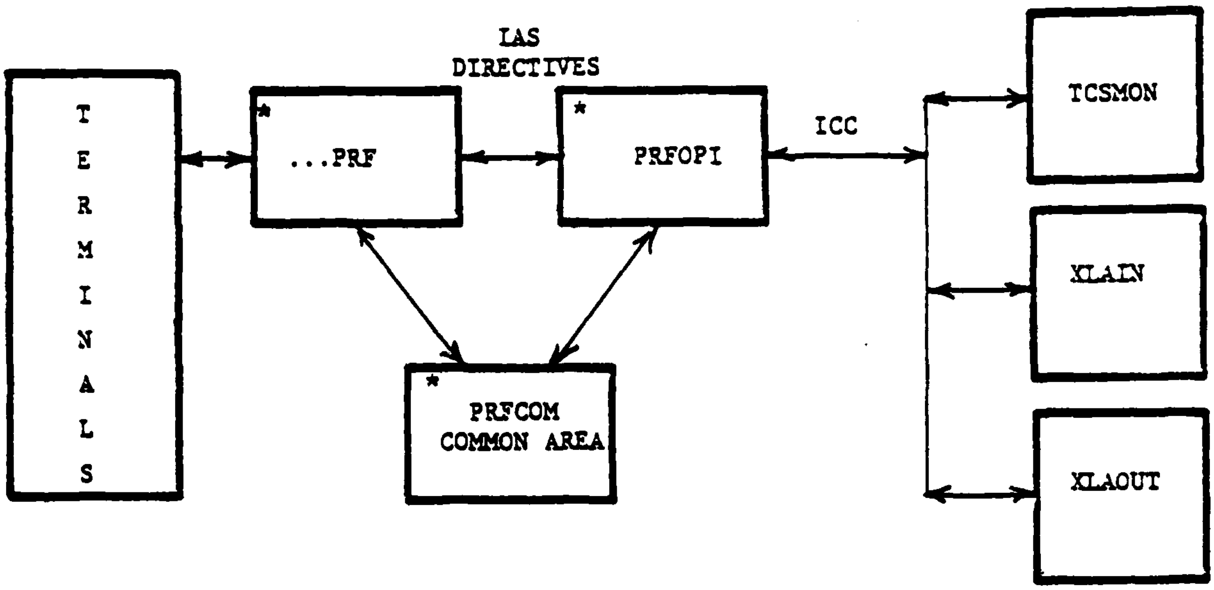


Figure 18. INTERACTIONS FOR PROFILE MAINTENANCE

Figure 19 shows an example of a typical analyst profile. Information following the colons (:), as shown in the example, is provided by the analyst in response to prompts from system software.

2.3.11 Soft Shutdown. MSS has the capability to initiate a soft shutdown of its own processes. This soft shutdown enables MSSGTY to continue processing on the SSB side while MSS ceases processing for trouble-shooting by the operator. While MSS is shutdown, MSSGTY will not send queries or messages but will simply allow files to collect.

A further feature of the MSS soft shutdown is the ability to separately shutdown messages or queries, allowing the on-going process to continue normally. A total shutdown is also possible. Figure 20 illustrates the transactions involved.

In addition to an MSS-invoked soft shutdown, MSSGTY can request a full or partial soft shutdown of MSS. If MSSGTY senses a problem within the MSS processes, a request for a soft shutdown will be issued. Normally, a notification of such a shutdown would be the response. The soft shutdown request transactions are depicted in Figure 21.

2.3.12 Restart. Restart of MSS processes under soft shutdown is the responsibility of MSS. MSS can notify SSB of a fatal restart of MSS functions or may limit restart to queries or messages only. Figure 22 shows the transactions involved.

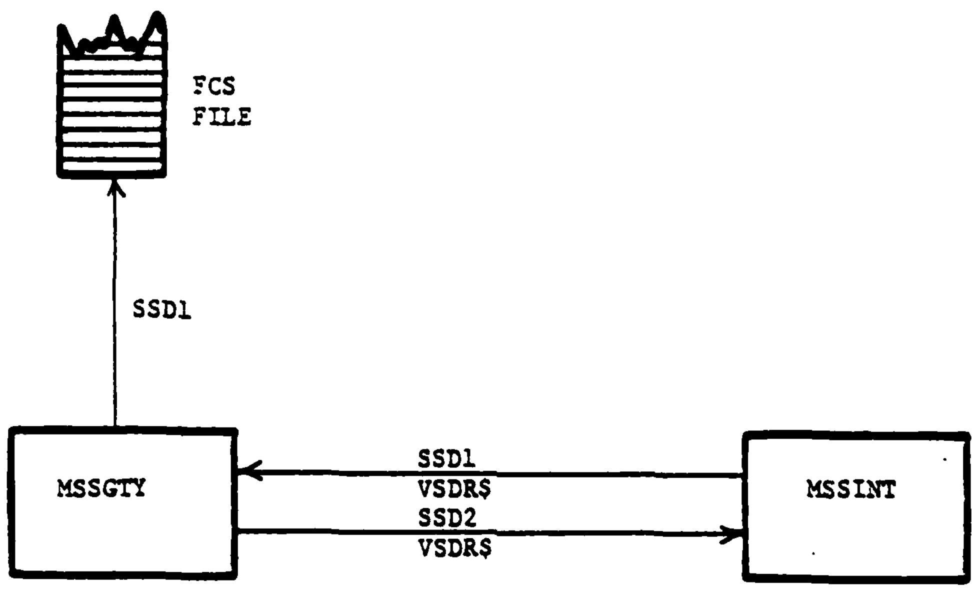
2.4 The Review Function. This function is part of a new concept in managing the flow of message traffic within the SSB system and of making the system both faster and easier to use. The queue manager concept represents a major revision to the SSB system. Virtually every program and user function is affected by the implementation of this new concept. It imposes a new query structure on the system and provides a function key capability to operate many of the SSB user functions. An overview of the design and operation of the queue manager concept and the changes affecting user operation is presented in the following paragraphs.

a. Queue Organization. Under the queue manager concept, message traffic entering the SSB system via external networks (e.g., AUTODIN) may be routed to a specified "watch officer" queue, or to subject areas redefined to the system via the MSS profile routing system. Each subject area contains two queues upon which message entries are stored; a Review queue and a Pending Action queue. A third type of queue called the Hold queue, is not subject-oriented but assigned to each UID when the system is installed at a site.

The Review queue serves as the input queue for its own subject area. Messages are placed on this queue automatically whenever messages are routed (or transferred) to the subject area the queue supports.

```
PROFILE NUMBER:  123

MESSAGE TYPE:    SI

USER ID:  DFLT

"PROFILE IDENTIFICATION STATEMENT.
    PROFILE NAME:  SOVIET AIRCRAFT.

"PROFILE LOGIC SECTION.
    LOGIC: A
           AND B.
"PROFILE ELEMENT DEFINITION SECTION.
    DEFINE: A:

      ZONE = SUB, TXT
      SYNONYMS = SOVIET, USSR:
        B:
      ZONE = SUB, TXT
      SYNONYMS = AIRCRAFT.

PROFILE DISSEMINEE ASSIGNMENT SECTION.
   DISSEM:  CRT:  SAIR;
            HC:  8009.

"PROFILE SUBJECT ASSIGNMENT STATEMENT.
   SUBJECT:  USSR, AIRCRAFT.

"END OF PROFILE
```

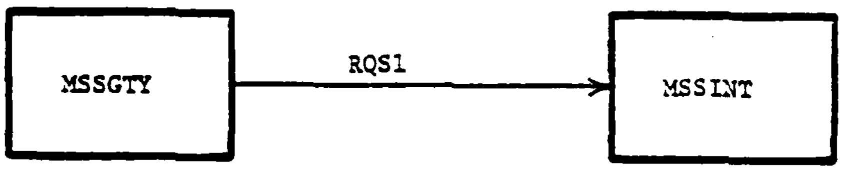
Figure 19.   **Example of Profile**



SSD1 - MSSINT notifies MSSGTY of full or partial shutdown. MSSGTY ceases transmission of affected files.

SSD2 - MSSGTY ACKs reception of shutdown.

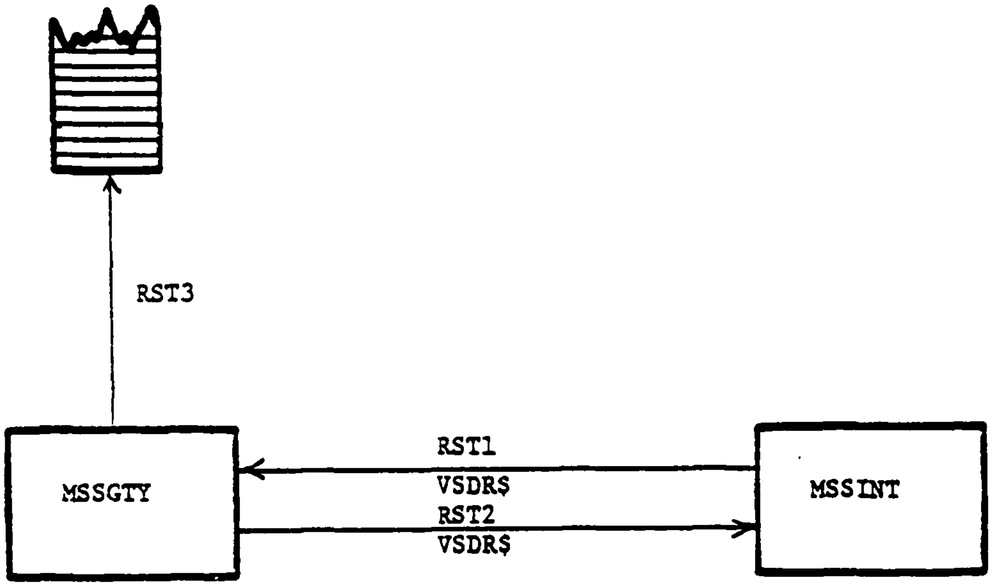
Figure 20. Soft Shutdown Initiation



RQS1 - MSSGTY senses problem and requests soft shutdown.

Response is either shutdown notification or a legitimate call for action.

Figure 21. Soft Shutdown Request



RST1 - MSSINT notifies MSSGTY of intent to resume function.

RST2 - Acknowledgment of restart.

RST3 - MSSGTY prepares to restart all affected nodes.

Figure 22. Restart of MSS

The Pending Action queue is the counterpart to the Output queue of previous editions of SSB. Messages are automatically placed on this queue as they are built, edited, or retrieved via the history function. Messages may also be placed on the queue by transfers from other queues.

The Hold queue has no operational counterpart in SSB. It was designed as a private work area for each UID assigned to the system. It is not a subject-oriented queue and is intended for use by the individual whom it is assigned. Messages may be transformed to this queue from the other queues of the subject area the user is signed on to.

b. Queue Manager Concept. The revised queue structure has a greater storage capacity than the input/output queue structure in previous versions of SSB. Each queue in the new system can accommodate up to 1280 messages entries; the old queues could accommodate only 32 entries each.

c. Function Key Operation. The capability to operate certain SSB user functions through use of function keys eases the burden of operation placed on the user/analyst by earlier versions of the system. Two types of terminals are currently supported for use with these function keys; the UNIVAC 1652, and the IBM 3270. Teletype terminals must still enter commands on the system and pseudo command lines of the terminal screen. The 1652 terminal has an advantage over the 3270 in that it has more function keys for use, and has the capability for illuminating keys which are valid for use and turning off those keys invalid for use at specific points during the operation of a given function. Figure 23 lists the function keys available for use and describes their purpose.

d. Data Structures.

1) Queue File. A queue file was created to maintain information concerning subject area names, UID access to those areas, and the messages stored on each queue in the system. The format of this file is illustrated in Figure 24.

2) Review Buffers and Paging Table. A review buffer was created to store records to be flashed to the screen. These buffers can accommodate seventeen 80-character lines, the size of the terminal display screen. A modified global routine reads records into the buffer until it is full. It is then flashed to the screen.

3) Audit Trails. An FCS file, TRAIL.FIL was created in ASCII format to store audit trail information. The file contains fields for precedence and security information, message sequence numbers, date-time-group, network or origin and/or message originator.

4) Distribution Lists. Provisions were made to create lists of subject areas to which messages may be transferred. Distribution lists are contained in a sequentially accessed TISFIL subfile labeled

FUNCTION	1652	3270	COMMAND	QUEUE	DEFAULT MESSAGE DISPOSITION
INITIALIZE TERMINAL	INIT & LFINIT	CLEAR	NONE	NONE	N/A
GET NEXT MESSAGE FROM REVIEW QUEUE	2	5	NXT	RVW	DEL
GET NEXT MESSAGE FROM PENDING ACTION QUEUE	3	6	NXT	PAQ	SAV
GET NEXT MESSAGE FROM HOLD QUEUE	4	7	NXT	HLD	SAV
DELETE CURRENT OR SPECIFIED MESSAGE	5	FNA	DEL	CURRENT QUEUE	DEPENDENT ON QUEUE
TRANSFER CURRENT MESSAGE TO PENDING ACTION QUEUE	7	PA1	TRF	PAQ	NO CHANGE
TRANSFER CURRENT MESSAGE TO HOLD QUEUE	8	PA2	TRF	HLD	NO CHANGE
TRANSFER CURRENT MESSAGE TO SPECIFIED SUBJECT AREAS	6	PF12	XFR	DEFAULTS TO RVW Q OF TARGET SUB-JECT	NO CHANGE
RETURN TO TRANSFER MESSAGE DISPLAY PAGE	9	FNA	TPG	CURRENT	NO CHANGE
REDISPLAY FIRST PAGE OF MESSAGE TEXT	10	FNA	FPG	CURRENT	NO CHANGE

N/A - Not applicable
FNA - Function not available

Figure 23. Function Key Assignments

SUBJECT AREA NAMES	MAXIMUM 64
UIDs WITH SUBJECT AREA ACCESS	
LIST OF MESSAGES ON REVIEW AND PROVIDING ACTION QUEUES FOR EACH SUBJECT AREA DEFINED TO THE SYSTEM	MAXIMUM 320 QUEUES
LIST OF MESSAGES ON THE HOLD QUEUE OF EACH UID DEFINED TO THE SYSTEM	
MESSAGE NAME	ONE ENTRY FOR EACH MESSAGE IN THE SYSTEM
SECURITY CLASSIFICATION DATA	
CURRENT REVIEW Q BIT MAP	IDENTIFIES WHICH REVIEW QUEUES THE MESSAGE IS ON
CURRENT PAQ BIT MAP	IDENTIFIES WHICH PENDING ACTION QUEUES THE MESSAGE IS ON
CURRENT HOLD Q BIT MAP	IDENTIFIES WHICH HOLD QUEUES THE MESSAGE IS ON
HISTORICAL REVIEW Q BIT MAP	IDENTIFIES THE QUEUES TO WHICH THE MESSAGE HAS BEEN ROUTED (INCLUDES CURRENT QUEUE)
MESSAGE DTG	
MESSAGE TRANSMISSION FLAG	

Figure 24. Queue File Format

and a Four-word bit map. The bit map corresponds to the four-word bit.

2.5 Queue Maintenance. Utility routines were developed to install, change, or remove subject area queues from the system, and to assign or revoke user access to the queues.

a. Installing Subject Areas. The INSUBJ routine is used to install new subject areas. When called, the routine issues a prompt to specify a subject as shown below (user entries are underlined):

MCR>RUN[6,3]INSUBJ$

SUB>XXXXXX — where XXXXXX is a six character alphanumeric name.

SUB>XXXXXX

.

.

.

SUB>CZ (conrol z terminates the routine)

The new subject areas are created with a Review and Pending Action queue each of which is assigned ten disk blocks (2560 words) which can accommodate 1280 message entries.

At present, 64 subject areas can be installed on the system. Error detection subroutines are included in INSUBJ to advise the user of duplicate names and when the maximum of 64 names is reached.

b. Changing Subject Area Names. The names of existing subject areas may be changed by running the CHGSUB routine. The Review and Pending Action queues of the old subject area are retained intact under the new name. The sequence of operation is illustrated below: (User entries are underlined).

MCR>RUN [6,3]CHGSUB$

OLD SUBJECT>XXXXXX — where XXXXXX and YYYYYY one each six-character alphanumeric names

NEW SUBJECT>YYYYYY

.

.

.

OLD SUBJECT>CZ

c. Removing Subject Areas. The REMSUB routine was provided to permit removing subject areas from the system. This routine also deletes the Review and Pending Action queues when the subject area is removed. The messages stored on the queues are still accessible for retrieval from disk by either the Review or History functions.

d. Assigning Subject Area Access. Users may be granted access to one or more subject areas by running the ANS2U routine. This routine

prompts a user to enter a subject name and to identify the UFOs permitted to access it. A sample assignment is shown below.

MCR>RUN ANS2U$ >>USAFR1 1,2,3,4,5 >>EOB 1,2,3,4,5 >>AOB 1,2 >>EOB 6,7,9,13	where USAFR1, EOB, and AOB are, individual characters, and the numbers identify the UIDs assigned access to the areas.

There is no limit to the number of UIDs that may be awarded access to a given subject area. If more than one program line is acquired to list all UIDs, the subject name may be enforced again to specify the additional UIDs. This case is illustrated in the sample showing two entries for the EOB subject area.

e. Revoking Subject Area Access. A user's permission to access a given subject area may be revoked by running the REMS2U routine. The format for this routine is the same as that for assigning access. A sample of the dialog required to run the routine is shown below:

```
MCR>RUN [6,3]REMS2U$
>> USAFE 5,9
>> AIROB 2,6
```

In this sample, access for UIDs 5 and 9 would have been revoked on subject area USAFE, and access for UIDs 2 and 6 from AIROB. The status of the messages contained in the subject areas unaffected; they will be accessible to any users still authorized to access the areas.

2.6 Unified Sign-On Function. System sign-on and sign-off functions were incorporated into a unified SIGNON program that replaces the separate LOGON and LOGOFF prorgrams featured in previous versions of SSB. The new program controls user and terminal access to three subsystems: Computer Assisted Tactical Intelligence System (CATIS), storage and retrieval processing system (SARP), and the Standard Software Base.

a. System Access.

User and terminal access authorization for each subsystem is maintained in a user identification file (USRFIL) and a terminal file (TEDFIL) respectively. Entries made by the user when he first attempts to sign-on are checked against the entries in these files to determine what subsystems the user and the terminal he is using are cleared to access. In addition, the security classification and overlay attributes of both the user and his terminal are also checked at the same time. The format of USRFIL is shown in Figure 25; that are TEDFIL in Figure 26.

b. Operating Procedures

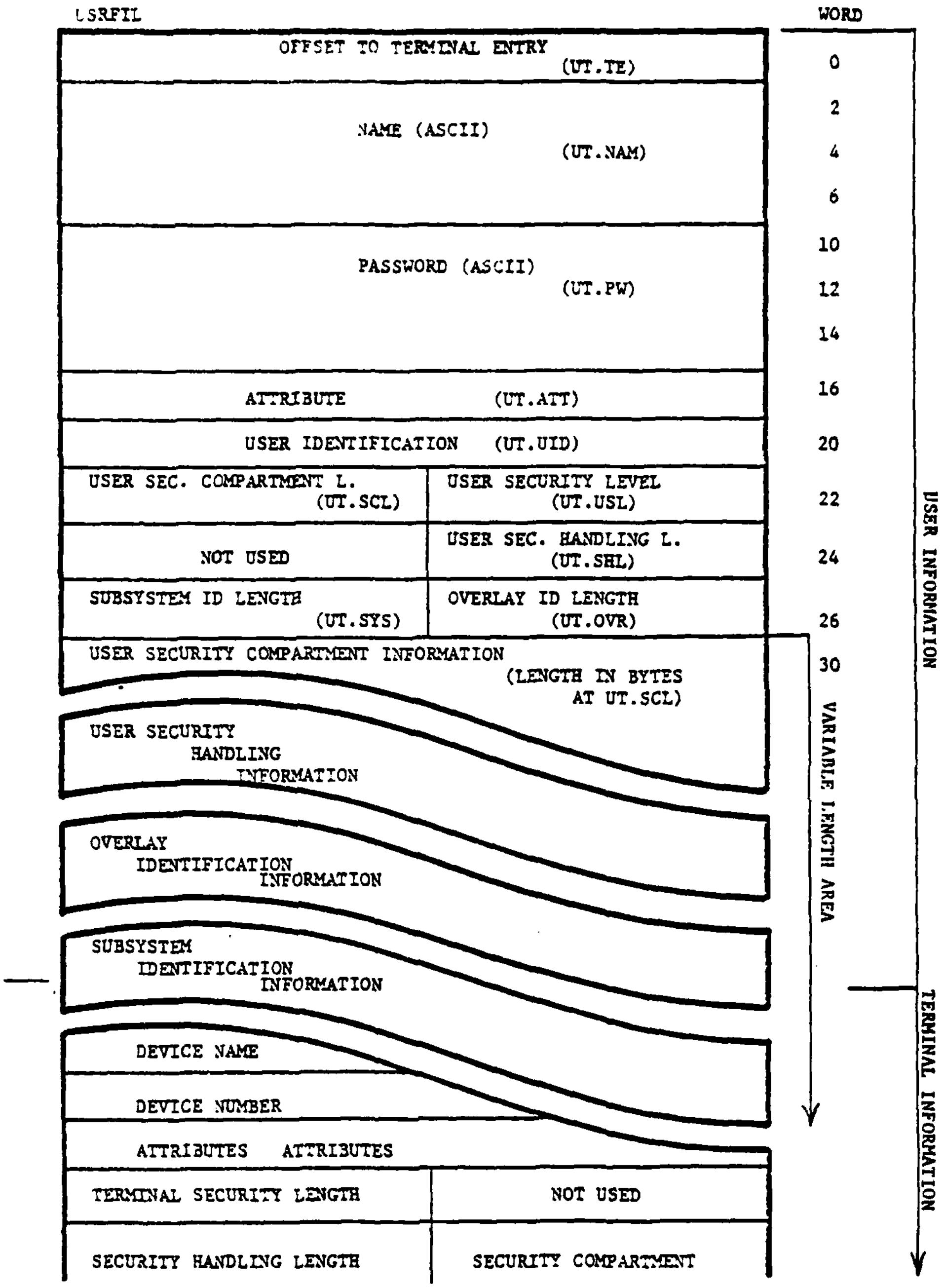


Figure 25. User File (USRFIL)

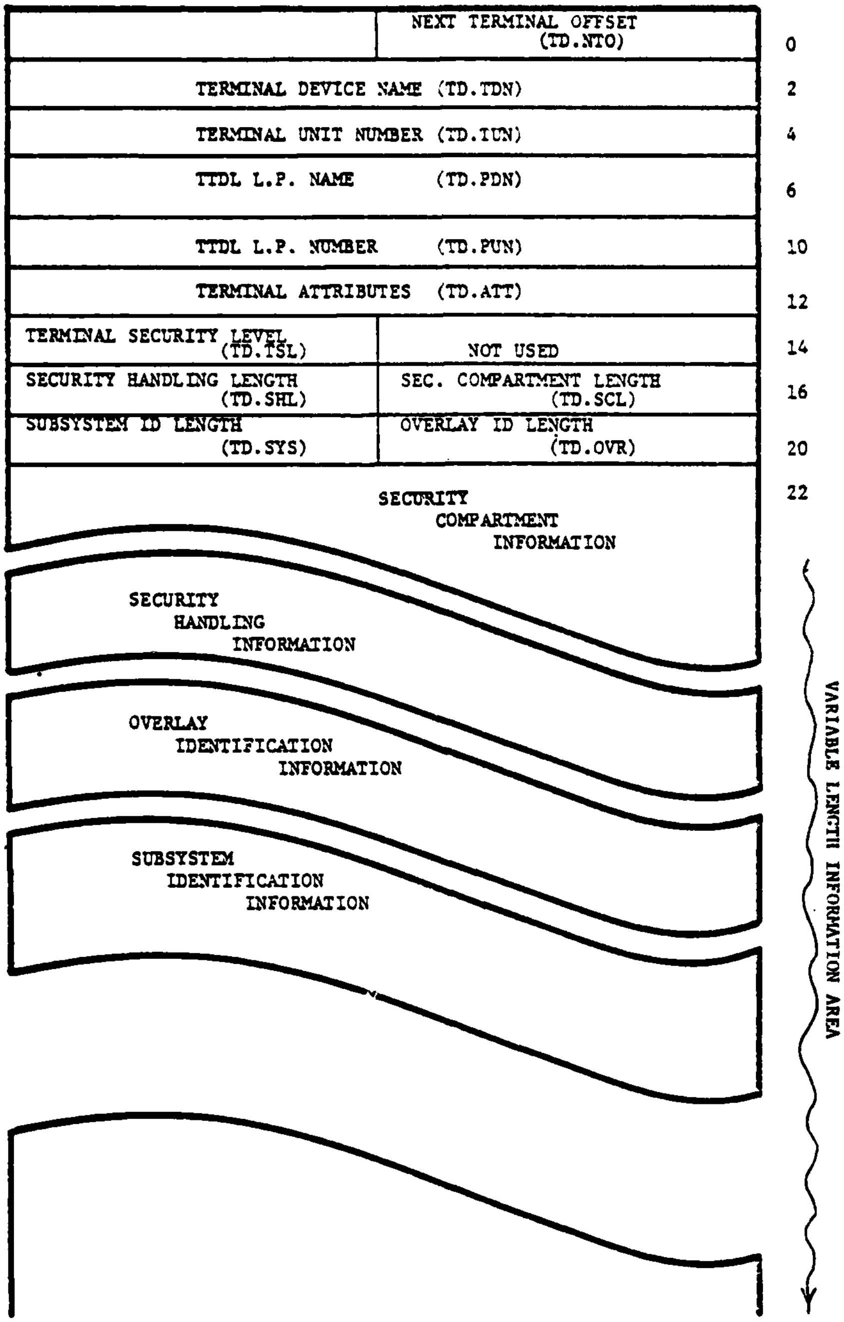


Figure 26. Terminal File (TEUFIL)

To the user, SIGNON first appears as a two-step process with a third step manifested when the SSB system is selected. In order to select any subsystem the user must first sign-on with the name, user identification, and password assigned to him and stored in the USRFIL. These entries are validated before any further action may be taken. As with earlier access programs, three unsuccessful attempts result in the user being denied access and required to reinitiate the program.

Following a successful sign-on a subsystem selection menu is displayed on the user's terminal. The menu lists only those subsystems that the user and the terminal are authorized to access. Subsystem access, as well as security classification ratings and overlay attributes for both the user and the terminal (CATIS) are specified by the entries in the USRFIL.

Selection of a subsystem places the terminal into an active state on the selected subsystem. The user must then use the operating procedures and protocol of that subsystem. When SSB is selected, a subject area menu is displayed at the terminal. As is the case with the subsystem menu, only those subject areas the user is authorized to use will be displayed. In the event the user is only authorized to access one subject area, no menu will be displayed and an advisory message will be output to the terminal identifying the subject area being accessed. Once a subject area has been selected, all SSB functions are available for use. All messages created by a user are sto. .d on the Pending Action queue of the selected subject area.

A user may change from one subject area to another, from one subsystem to another, or terminate the terminal session. To access another subject area the subject area menu must be called up either via a designated function key (for 1652 terminals) or by the keyword SUBJECT on the system command line of all supported terminals, e.g., 1652, 3270, and TTYs. To change to another subsystem or to terminate the terminal session the SIGNOFF function is used via function key or keyword command. When the subsystem menu appears, the user may select a subsystem or exit from the terminal session. If the user exits, the signon function must be reinitiated in order to access any system again.

2.7 Installation and Site Support

2.7.1 Military Airlift Command (MAC).

1. A troubleshooting trip was conducted in February 1980 to resolve problems with the Display Function at the Scott AFB SSB site. The function frequently terminated processing in the middle of a message with TTDL set field errors, and occasionally failed to come out of the suspended state while awaiting for a flash to the terminal screen to be completed. This problem caused the terminal to fail to respond to further display function inputs but did not affect other application program operations. While these problems were being analyzed. TTDL ran out of nodes and the system died. A patch to the UNIVAC spooler was

phoned in to eliminate the node problem and an INCO maintenance team visited the site to take care of the problems with the Display Function. A patched version of the display program was installed on a temporary UIC, and the problems did not reoccur. The patched program was then installed under the configuration management UICs and a tape was cut to record the changes.

In March, SSB Release 3.1 was installed at the MAC site. This version of SSB contains the revised queue structure and the new Review Function. The new release was successfully brought up using both the RSX-11D Version 6.2 and IAS Version 3.0 DEC operating systems.

2.7.2 USAFE.

1. The revised queue structure and the Review Function were successfully installed at USAFE under IAS Version 3.0 during the latter part of April 1980.

2. The site was revisited in August-September 1980 to satisfy the following objectives:

- Establish a working interface to transfer AUTODIN messages from the SSB computer to the local IBM 370 performed jointly with Aerospace personnel.

- Resolve problems in the operational SSB system
- Assist on-site personnel in the installation of UNIVAC 1652 terminals on the SSB sytem and making them operational

- Provide on-site training to USAFE personnel in the use of the 1652 terminals, and advise personnel of the latest improvements to SSB

The trip was only partially successful. System operation using the 1652 terminals was much more reliable than with the 3277 terminals used when SSB Release 3.1 was installed. Aerospace software to interface with the SSB system was not ready soon enough to start constructive integration testing.

SECTION 3. TECHNICAL DOCUMENTATION

The technical documentation provided during the contract period consists of a base line documentation series defined in the Contract Data Requirements List (CDRL), and other documentation related either to specific software deliverables specified in the SOW, or to ad hoc requirements generated by the delivery and implementation of interim operating capabilities.

3.1 CDRL Line Items

A001 - R&D Status Reports

1 - 25 Sep 79 - 25 Oct 79
2 - 26 Oct 79 - 25 Nov 79
3 - 26 Nov 79 - 25 Dec 79
4 - 26 Dec 79 - 25 Jan 80
5 - 26 Jan 80 - 25 Feb 80
6 - 26 Feb 80 - 25 Mar 80
7 - 26 Mar 80 - 25 Apr 80
8 - 26 Apr 80 - 25 May 80
9 - 26 May 80 - 25 Jun 80
10 - 26 Jun 80 - 25 Jul 80
11 - 26 Jul 80 - 25 Aug 80
12 - 26 Aug 80 - 25 Sep 80

A002 - Test and implementation Plan, September 1980

A003 - Standard Software Base (SSB) Overview Document, September 1980

A004 - SSB Program/System/Subsystem Installation and Maintenance Manual

Part A, Program Specifications, September 1980

Part B, Installation and Maintenance Manual, September 1980

A005 User, Operator, and Programmer Manual:

Part A, SSS User's Manual, September 1980

Part B, SSB Computer Operator's Manual, September 1980

Part C, SSB Programmer's Manual, September 1980.

A007 Final Technical Report, September 1980.

3.2 Other Requirements

SSB/DSSCS Test and Implementation Plan, November 1979 (SOW 4.1.1.2)

SSB/GENSER Test and Impementation Plan (incorporated into CDRL A002)

SSB/MSS Test and Impementation Plan, September 1980

Interim Operating Instructions, Review Function, May 1980

Program Specifications for the Messsage Support Subsystem (MSS) Software (incorporated into A004) September 1980

SSB/TTDL Configuration Management Plan, April 1980

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

LMED

8